

#12/Reply Brief  
T. McBeth Brown  
3/23/02

COPY OF PAPERS  
ORIGINALLY FILED

PATENT

Docket No. EN995139



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: G. W. Wilhelm, Jr.

Application No.: 08/820,181

Group No.: 2151

Filed: 14 March 1997

Examiner: M. Banahkhah

For: System and Method for Queue-less Enforcement of Queue-like Behavior On Multiple Threads Accessing a Scarce Resource

RECEIVED

MAR 22 2002

Assistant Commissioner for Patents  
Washington, D.C. 20231

Technology Center 2100

ATTENTION: Board of Patent Appeals and Interferences

REPLY BRIEF (37 C.F.R. § 1.193(b))

Sir:

This is a Reply Brief to the Examiner's Answer dated Jan. 2, 2002, paper No. 11.

Grouping of Claims

In the Examiner's Answer of Jan. 2, 2002, appellant is first advised of the allowance of Claim 8.

Appellant requests that Claim 3 be grouped together with claim 8. Claim 3 is substantially similar to claim 8, differing in that the preamble of claim 8 casts the claim in the style of a Beauregard claim.

**The Davidson and Periwal References Do Not Teach or Achieve  
Fairness as Defined in Appellant's Claims**

Standard OS multi-tasking and serialization primitives, such as those used by Periwal, do nothing to ensure FIFO (fair) access to a resource by multiple requesters. All threads waiting for access to a resource are all made runnable when the resource is released via a mutex or similar kernel primitive operation. ONLY ONE can then gain access to that resource (in no guaranteed order), and all others must be put to sleep again. The actions of making a thread runnable and then putting it back to sleep are kernel operations that consume CPU resource. The resulting thrashing of the run queue inside the kernel is a major cause for poor performance in multi-threaded applications.

Appellant's claims all recite a system or method for assuring fair, that is FIFO, order among threads and various mechanisms (such as a stationary queue or counter pair) for

achieving that result. The concept of fairness is variously claimed as follows:

Claim 1	one-by-one in order of request
Claims 2,3	in FIFO order
Claims 4-7	in fair order
Claims 3,8	next...in line

The Examiner observes (Examiner's Answer, page 4) that Davidson does not "explicitly teach a stationary queue for allocating access to the resources one-by-one in order of request". Appellants agree. However, the Examiner then seems to argue (Answer, pages 4-5) that Periwai teaches such at Col. 3, lines 45-57. This is a new point of argument, and one which appellant respectfully traverses. Periwai states:

"When a new mutex is acquired from the operating system, that "real" mutex (or a handle or a pointer to it) is stored in the "real" mutex field of the surrogate mutex (i.e., the mutex record); the mutex ID is set equal to the ID for the thread or process achieving acquisition of the mutex; and the mutex count is set equal to one. Later calls to acquire a mutex for a resource, such as those that may occur in a nested fashion, are processed by first checking the ID of the thread or process (currently requesting the mutex) against the mutex ID stored in the mutex record. In the event that the two IDs match, the system simply increments the mutex reference count and permits the thread or process to continue execution." (Periwai, col. 3, lines 45-57).

Periwai goes on, however, to explain what this is all

about, in the following paragraph:

"In a corresponding manner, when a thread or process releases the (surrogate) mutex, the system of the present invention decrements the mutex reference count. The "real" mutex is not released at the level of the operating system, unless the mutex reference count has reached zero. Thus, a thread or process can effectively acquire and release the mutex multiple times. Regardless of how many times the mutex is acquired or released, the thread or process will retain acquisition of the mutex as long as the number of requests for acquisition exceeds the number of releases for the mutex. When the reference count reaches zero, then the system can invoke the corresponding operating system service of releasing the "real" mutex which was held (whereupon the next waiting thread or process acquires the mutex and proceeds with its processing)." (Periwall col. 3 line 58 to col. 4 line 5.)

Periwall characterize this as "nested acquisitions of mutexes". (Periwall, col. 4, lines 7-8.)

#### **Periwal's Nested Mutex Does Not Teach Fairness**

Periwal's teaching of a mutex ID stored within the mutex record is NOT for enforcement of fair or FIFO order. It is simply to prevent a deadlock caused by having the same thread request (again) a mutex that it already holds. In such case a reference count is incremented. The reference count means that there must be as many releases of the mutex by the holder as there are acquisitions of the mutex by that same holder before the resource can really be considered

"free". If a thread were to attempt to acquire (again) a mutex it already holds, without such a reference count and mutex ID check, that same thread would be put to sleep awaiting the release of a mutex it already owned: this is a deadlock. The Periwal mechanism serves to protect against bad coding practice (acquiring the same resource multiple times within the same thread) but does nothing to enforce fair or FIFO order dispatching of a resource among multiple waiting requesters once the thread using the resource has done all its releases and is finished with the resource.

No mention is ever made in Periwal as to how fair order is maintained, and the use of mutex technology inherently results in undefined (that is, not fair) order, as Appellant discusses in his Brief.

With respect to claim 5, the Examiner correctly observes (Answer, pages 6-7) that Periwal maintains a counter. He incorrectly asserts that it maintains "a number of threads which have been forced to wait and have been subsequently satisfied". Instead, the Periwal counter maintains track of the number of excessive access requests by the same thread, thus enabling the surrogate mutex to prevent deadlock and from releasing the resource before the thread using it is completely finished.

The Examiner asserts that appellant's specification does not teach FIFO order. (Answer, page 9-10). The Examiner seems to assert that if a thread "waiting in the stationary queue the longest" is the next "made ready to run by the operating system" that thread is not serviced in FIFO order. Appellant traverses. That is exactly FIFO. The next thread to go is by every possible way of thinking the one that has been there the longest. This is exactly what is enforced by appellant's counter-pair or stationary queue and is precisely what Appellant describes in his specification at page 9, Table 2.

**Appellant Properly Describes and Claims a Unique and  
Unobvious Stationary Queue or Counter Pair Which Provides  
Fairness to Waiting Threads**

At page 8, the Examiner asserts that appellant's invention does not provide a "stationary queue". Apparently based on personal knowledge, he states that "For ... [a person of] ordinary skill in the art a 'stationary queue' is a queue, wherein, when data is entered it does not pop out conditionally" and, further, "For a person [of] ordinary skill in the art, a stationary queue is one where data are in and out serially."

The Examiner has provided no reference in the art which uses the term "stationary queue", certainly not in the in the manner defined and used by Appellant. This assertion has not been made previously by the Examiner, and Appellant has not been afforded the opportunity, as is his right under 37 C.F.R. Section 1.1107(b), to request an affidavit from the Examiner which would be subject to contradiction or explanation by the affidavits of Appellant and other persons. See In re Maloy, 328 F.2d 933, 140 USPQ 599, 601 (C.C.P.A. 1964).

Applicant has used the term "stationary queue" specifically to describe structure or steps which achieve queue-like behavior (that is, fairness) without being a queue. This is discussed at pages 14-15 of Appellant's Brief.

With respect to claims 1-3, where using a term (in this case, "queue") is used in a manner other than its ordinary meaning (in this case, to refer to a structure in which queued data or identifiers are not moved), applicant must provide a definition of the term in the specification or prosecution history. (See In re Morris, 127 F.3d 1048, 44 USPQ 2d 1023 (Fed. Cir. 1997). Applicant has done so by defining his counter structure 110, 112 as a "stationary

S/N 08/820,181 7 EN995139

queue". The specification can be relied upon to assist in ascertaining the scope and meaning of the language employed in the claims, (United States v. Teletronics Inc., 857 F.2d 778, 8 USPQ 2d 1217, 1220 (Fed. Cir. 1988)) and a claim is to be interpreted in accordance with a specific definition set forth in the specification (Beachcombers, Int'l, Inc. v. WildeWood Creative Products Inc., 31 F.3d 1154, 31 USPQ 2d 1653 (Fed. Cir. 1994)).

Claim 2 recites specific structure (sleep code for generating block ID and wake-up code for generating run ID) for implementing the counters which are the stationary queue. The Examiner asserts that such is taught by Periwal. (Answer, p. 8.) However, Periwal's counters are not used in a manner to implement a stationary queue which, as claim 2 explicitly requires, grants access to a resource in FIFO order.

Appellant variously claims the concept of "fairness" to distinguish the art and provide explicit limitations describing the operation of the "stationary queue" in those claims on appeal which recite it. As previously discussed in his Brief (at page 9), appellant's stationary queue is described and claimed (in claims 1-3) as one which assures "fair order" or "in order of request".



Further, apparently based on personal knowledge, the Examiner states that regarding "the word 'fairness', the word is broad with no specific [meaning] and it is subject to different interpretation by a person of ordinary skill in the art." (Answer, page 9.) This is the first time the Examiner has made this assertion, and Appellant has not heretofore been given the opportunity to exercise his right under 37 C.F.R. Section 1.1107(b) to request the affidavit of the Examiner and respond with contradictory or explanatory affidavits.

The Examiner appears to struggle with what the word "fair" means in the context of the present invention. If a person stands in line at a supermarket, waiting to check out in turn, or if a thread is waiting on a resource, the definition of fairness is pretty obvious, certainly so to those of ordinary skill in the art. Priority order (which the Examiner suggests teaches "fairness") is not "fair" -- those of ordinary skill in the art would understand that "fair" is FIFO, and that "priority" order is not FIFO and is not "fair".

## Conclusion

Appellant requests that the rejection of claims 1-7 be reversed and the case remanded for issuance with claims 1-8.

Shelley M Beckstrand 3-1-02

Shelley M Beckstrand  
Attorney for Appellant  
Reg. No. 24,886

314 Main Street  
Owego, NY 13827

Phone: (607) 687-9913  
Fax: (607) 687-7848